



## **Govern de les Illes Balears**

Conselleria d'Economia, Hisenda i Innovació  
Direcció General de Tecnologia i Comunicacions

# Estándar de desarrollo de aplicaciones del Govern de les Illes Balears

## **BASE DE DATOS**

Versión 5.0

Fecha Revisión: 28/05/09

# Índice de contenidos

<b>INTRODUCCIÓN.....</b>	<b>3</b>
<b>NOMENCLATURA DE BASE DE DATOS.....</b>	<b>4</b>
2.1. CONSIDERACIONES GENERALES .....	4
2.2. NOMENCLATURA DE TABLESPACES .....	4
2.3. NOMENCLATURA DE TABLAS Y VISTAS.....	5
2.3.1. Nomenclatura de tablas con campos LOB .....	5
2.3.2. Nomenclatura de columnas .....	6
2.4. NOMENCLATURA DE SECUENCIAS .....	6
2.5. NOMENCLATURA DE DISPARADORES ( <i>TRIGGERS</i> ).....	6
2.6. NOMENCLATURA DE RESTRICCIONES ( <i>CONSTRAINTS</i> ) .....	6
2.6.1 Clave primaria .....	6
2.6.2. Claves extranjeras .....	6
2.6.3. Otras restricciones .....	7
2.7. NOMENCLATURA DE ÍNDICES .....	7
2.8. NOMENCLATURA DE ROLES .....	7
2.9. NOMENCLATURA DE PROCEDIMIENTOS, FUNCIONES, PAQUETES Y RESTO DE OBJETOS .....	8
2.10. NORMAS REFERENTES A SINÓNIMOS.....	8
2.11. ACCESO A LA BASE DE DATOS. NORMAS REFERENTES A LOS PRIVILEGIOS DE ACCESO .....	8
2.12. RESTRICCIONES ADICIONALES Y RECOMENDACIONES .....	9
<b>PROBLEMAS FRECUENTES.....</b>	<b>10</b>

# Capítulo 1:

## Introducción

Este documento detalla el estándar de base de datos que se debe seguir para el desarrollo de aplicaciones que se instalarán en los servidores de la DGTIC.

Las principales diferencias respecto de la versión anterior del documento, son las siguientes:

- Se aumenta la longitud de los nombres de objetos (antes sólo podían tener 6 letras, además del prefijo) a 30 caracteres.
- Se detalla el estándar para la utilización de objetos LOB.
- Se detallan los problemas más frecuentes a la hora de ejecutar los *scripts* en la base de datos.

# Capítulo 2:

## Nomenclatura de Base de datos

### 2.1. Consideraciones generales

Las aplicaciones deberán funcionar sobre bases de datos Oracle 9.2.0.5 y estar preparadas para funcionar en Oracle 10.2.0.4 sin tener que realizar ningún cambio.

El código de aplicación y su prefijo habrán sido facilitados previamente por el *Centre de Procés de Dades* de la DGTIC (ver documento de Implantación de aplicaciones, Capítulo 2. Solicitud de código de aplicación).

Todos los objetos de base de datos de una aplicación serán propiedad de un mismo usuario de base de datos, que deberá coincidir con el código de aplicación previamente asignado por la DGTIC.

Asimismo, todos los objetos de base de datos empezarán por un prefijo de tres letras, también asignado por la DGTIC, seguidos de un guión bajo (\_).

En los ejemplos incluidos en este documento, cada vez que se haga referencia al código de aplicación se utilizará como ejemplo el literal '**APLICACION**', y como prefijo el literal '**APL**'.

### 2.2. Nomenclatura de tablespaces

En general, todos los objetos del usuario de base de datos se ubicaran en el *tablespace* de nombre igual que el usuario (*tablespace* por defecto del usuario).

Esto es válido para aplicaciones de tamaño reducido. Si la aplicación hace un uso intensivo de índices, es recomendable utilizar *tablespaces* separados para datos e índices. Si la aplicación contiene LOBs, por cuestiones de rendimiento y administración, es obligatorio almacenarlos en un *tablespace* separado del resto de objetos.

En el momento de solicitar el código de aplicación se tiene que indicar si la aplicación utiliza LOBs y si se quieren tener datos e índices separados.

En el caso de tener los objetos en dos o más *tablespaces* (datos, índices y LOBs) los nombres dependerán del tipo de aplicación, como se indica en la siguiente tabla.

Tipo de aplicación	Tablespace de datos	Tablespace de índices	Tablespace de LOB
Aplicación de tamaño reducido	APLICACION	APLICACION	No permitidos
Aplicación de tamaño reducido con LOB	APLICACION	APLICACION	APLICACION_LOB
Aplicación con datos e índices separados	APLICACION_DADES	APLICACION_INDEX	No permitidos
Aplicación con datos e índices separados y LOB	APLICACION_DADES	APLICACION_INDEX	APLICACION_LOB

Para separar los objetos en un mayor número de *tablespaces* se deberá consultar con los administradores de las bases de datos de la DGTIC.

## 2.3. Nomenclatura de tablas y vistas

Seguirán el patrón **APL\_XXX**.

Donde XXX es un nombre representativo de la entidad a la que corresponde.

Ejemplos:

APL\_CLIENT  
APL\_FACTURA

En las tablas resultantes de una relación N:M, el nombre de la tabla contendrá el nombre (o parte de él) de cada una de las tablas.

Ejemplo: tabla resultante de una relación N:M entre APL\_CLIENT y APL\_FACTURA:

APL\_CLIENT\_FACTURA

### 2.3.1. Nomenclatura de tablas con campos LOB

Para crear tablas que tengan algún campo de tipo LOB (BLOB, CLOB o NCLOB) se seguirá el siguiente formato (en el ejemplo se crea la tabla APL\_TABLA):

```
CREATE TABLE APL_TABLA
(
  campo1 tipo_campo1,
  campo2 CLOB,
  campo3 BLOB)
TABLESPACE APLICACION_DADES
  LOB (campo2) STORE AS APL_TABLA_campo2_LOB
    (TABLESPACE APLICACION_LOB
     INDEX APL_TABLA_campo2_LOB_I)
  LOB (campo3) STORE AS APL_TABLA_campo3_LOB
    (TABLESPACE APLICACION_LOB
     INDEX APL_TABLA_campo3_LOB_I);
```

donde,

APL_TABLA	representa el nombre de la tabla
campo2	representa un campo de tipo CLOB
campo3	representa un campo de tipo BLOB
APLICACION_DADES	representa el nombre del <i>tablespace</i> de DATOS
APLICACION_LOB	representa el nombre del <i>tablespace</i> de LOB

El nombre del objeto LOB (cláusula STORE AS) y de su índice asociado se forma añadiendo al nombre de la tabla el nombre del campo LOB, seguido del sufijo `_LOB` (o `_LOB_I` para el índice del LOB). Si el nombre resultante supera los 30 caracteres, se tiene que reducir el nombre del campo o de la tabla, pero siempre manteniendo el prefijo de la aplicación (APL\_) y el sufijo (`_LOB` o `_LOB_I`).

### **2.3.2. Nomenclatura de columnas**

Se eliminan todas las restricciones de nomenclatura de la anterior versión del estándar.

Los nombres de columna de cada tabla son libres, ya no tienen que empezar por las tres letras identificativas del nombre de la tabla, y tan solo tienen la limitación del tamaño máximo de 30 caracteres.

## **2.4. Nomenclatura de secuencias**

Seguirán al patrón **APL\_XXX\_SEQ**.

Donde XXX es un nombre representativo de la tabla o campo para la cual se crea la secuencia.

Ejemplo:

APL\_CLIENT\_SEQ: para la secuencia del código de la tabla APL\_CLIENT.

## **2.5. Nomenclatura de disparadores (*triggers*)**

Seguirán al patrón **APL\_XXX\_YYY\_TRG**

Donde XXX indica el nombre de la tabla a la que se asocia el disparador, y YYY es un nombre representativo del propio disparador.

Ejemplo:

APL\_CLIENT\_ALTA\_TRG

## **2.6. Nomenclatura de restricciones (*constraints*)**

### **2.6.1 Clave primaria**

Seguirán el patrón **APL\_XXX\_PK**

Donde XXX indica el nombre de la tabla para la cual se crea la clave primaria.

Ejemplo:

APL\_CLIENT\_PK

### **2.6.2. Claves extranjeras**

Seguirán al patrón **APL\_XXX\_YYY\_FK**

Donde XXX indica el nombre de la tabla de origen y YYY indica el nombre de la tabla referenciada.

Ejemplo: clave extranjera de la tabla APL\_CLIENT hacia la tabla APL\_ILLA

APL\_CLIENT\_ILLA\_FK

### **2.6.3. Otras restricciones**

Seguirán al patrón **APL\_XXX\_YYY\_ZZ**

Donde XXX indica el nombre de la tabla, YYY es un nombre representativo del campo o campos afectados y ZZ es el sufijo que indica lo que hace la restricción, que puede tomar uno de estos valores:

\_UK: para claves únicas (UNIQUE)

\_CK: para restricciones de comprobación (CHECK)

\_NN: para restricciones no nulas (NOT NULL).

Nota: para las restricciones NOT NULL no es necesario definir el nombre si se hacen en línea (cuando se define el nombre del campo en la sentencia de creación de tabla). Para el resto de restricciones sí que se tiene que definir el nombre siguiendo el patrón.

Ejemplos:

APL\_CLIENT\_NIF\_UK: clave única para el campo NIF de la tabla APL\_CLIENT

APL\_CLIENT\_SEXE\_CK: comprobación del campo SEXE de la tabla APL\_CLIENT

APL\_CLIENT\_EDAT\_NN: el campo EDAT de la tabla APL\_CLIENT no puede ser nulo

## **2.7. Nomenclatura de índices**

Seguirán al patrón **APL\_XXX\_YYY\_I**

En general, los índices siguen la misma nomenclatura que la *constraint* correspondiente, seguida del sufijo '\_I'.

Para el resto de índices, XXX indica el nombre de la tabla y YYY el nombre del campo a indexar.

Ejemplos:

APL\_CLIENT\_PK\_I: índice para la clave primaria

APL\_CLIENT\_ILLA\_FK\_I: índice para la clave extranjera

APL\_CLIENT\_NUM\_SS\_I: índice para el campo NUM\_SS

## **2.8. Nomenclatura de roles**

Seguirán al patrón **APL\_XXX**

Donde XXX es un nombre representativo del rol.

Ejemplos:

APL\_CONSULTA  
APL\_MANTENIMENT  
APL\_ADMINISTRACIO

## 2.9. Nomenclatura de procedimientos, funciones, paquetes y resto de objetos

En estos casos, la nomenclatura es más libre, siempre que se siga la norma de empezar cada nombre por el prefijo de la aplicación, y que el nombre del objeto sea el más simple y representativo posible.

Aunque no es obligatorio, se recomienda utilizar un sufijo para cada tipo de objeto a fin de identificarlos rápidamente. Se proponen los siguientes:

<i>Procedimientos</i>	APL_XXX_PR
<i>Funciones</i>	APL_XXX_FN
<i>Paquetes</i>	APL_XXX_PQ

## 2.10. Normas referentes a sinónimos

La utilización del prefijo particular de la aplicación hace que cada nombre de objeto sea único dentro de la base de datos. Eso permite que todos los objetos de cada aplicación tengan asignados los correspondientes sinónimos públicos. Es necesario adjuntar los *scripts* de creación de estos sinónimos públicos para las tablas, vistas, secuencias, procedimientos, funciones y paquetes de la aplicación.

Ejemplo:

```
CREATE PUBLIC SYNONYM APL_CLIENT FOR APLICACION.APL_CLIENT
```

Nota: En ningún caso se utilizará en los *scripts* de creación de sinónimos la opción CREATE OR REPLACE. Si se tiene que sustituir un sinónimo público, se tiene que hacer antes el DROP PUBLIC SYNONYM y luego volverlo a crear.

## 2.11. Acceso a la base de datos. Normas referentes a los privilegios de acceso

Para el acceso a la base de datos deberá definirse un *pool* de conexiones. El usuario del *pool* de conexiones seguirá la nomenclatura WWW\_APLICACION, donde APLICACION coincida con el código de aplicación asignado por la DGTIC.

Para que el usuario WWW\_APLICACION pueda utilizar los objetos del usuario propietario (usuario APLICACION) será necesario dar los privilegios de acceso (*grants*) adecuados. Los permisos no se darán directamente a usuarios finales. Se darán solamente a roles o a usuarios WWW\_APLICACION.

Los únicos permisos que se permiten asignar a roles o usuarios son:

```
SELECT, INSERT, UPDATE, DELETE, EXECUTE
```

Ejemplo: las sentencias GRANT relativas a la tabla APL\_CLIENT de la aplicación APLICACION podrían ser:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON APL_CLIENT TO  
WWW_APLICACION;
```

## 2.12. Restricciones adicionales y recomendaciones

- El nombre de los objetos de base de datos será como máximo de 30 caracteres, y sólo pueden incluir los caracteres A-Z, a-z , 0-9 y guión bajo (\_).
- La creación de los objetos no puede incluir comillas en la definición del nombre del objeto.
- No se permitirá la utilización de campos de tipo LONG.
- El juego de caracteres de las bases de datos es UTF8 (NLS\_CHARACTERSET = UTF8).
- El *national character set* es AL16UTF16 (NLS\_NCHAR\_CHARACTERSET = AL16UTF16).
- Las Bases de Datos tiene el valor: NLS\_LENGTH\_SEMANTICS = CHAR.
- En el caso de crear campos de tipo VARCHAR2 o CHAR, al indicar el número de caracteres se prestará atención a que no ponga "BYTE".
- Para evitar los abrazos mortales (*deadlocks*) que se producen al borrar registros de una tabla que tiene tablas relacionadas (tablas hijas), se recomienda crear un índice para las claves extranjeras de la tabla hija.

# Capítulo 3:

## Problemas frecuentes

Listado de los problemas más comunes detectados durante la validación de estándares de Base de datos, y que sería conveniente revisar antes de pedir una instalación de aplicaciones en los servidores de la DGTIC.

### 1.- Nomenclatura de objetos

El principal problema encontrado es el referente a la nomenclatura de objetos. En general se sigue el estándar de nomenclatura para el nombre de las tablas, pero para el resto de objetos no se cumple (índices, claves primarias y extranjeras, etc.).

### 2.- Objetos en *tablespaces* incorrectos

Cuando se crea un usuario con dos o más *tablespaces*, el *tablespace* por defecto del usuario es el de datos. Muchas veces se definen los índices sin indicar el nombre del *tablespace* de índices, por lo que se crea en el *tablespace* por defecto del usuario. A la hora de crear un índice o un LOB, se tiene que indicar expresamente que se cree en el *tablespace* correspondiente.

Ejemplo:

```
CREATE INDEX APL_CLIENT_NIF_I ON APL_CLIENT(NIF) TABLESPACE
APLICACION_INDEX;
```

### 3.- Uso de campos **BYTE** en lugar de **CHAR**

Las bases de datos de la DGTIC utilizan el NLS\_LENGTH\_SEMANTICS a CHAR. Si la empresa desarrolladora lo tiene definido en BYTES, a la hora de mandar los scripts de creación de tablas se envían con BYTE en la definición de columnas de tipo VARCHAR2. El uso de BYTE no está permitido por la DGTIC, se tiene que enviar en CHAR o sin poner nada.

Ejemplo:

```
Incorrecto CREATE TABLE APL_CLIENT
              (campo1 VARCHAR2(100 BYTE));
```

```
Correcto CREATE TABLE APL_CLIENT
              (campo1 VARCHAR2(100 CHAR));
```

### 4.-Asignación de privilegios incorrectos

Los únicos privilegios permitidos son SELECT, INSERT, UPDATE, DELETE y EXECUTE.

No puede haber privilegios ALTER ni REFERENCES. En caso de tener que utilizarlos se tiene que consultar antes con los administradores de base de datos de la DGTIC.

Es habitual encontrar en los scripts GRANT ALTER a secuencias. Si se tiene que modificar una secuencia, se tiene que hacer enviando el correspondiente cuaderno de carga a la DGTIC con la sentencia DDL correspondiente.

## 5.-Envío del cuaderno de carga y ejecución de scripts de base de datos

La ejecución de los scripts de base de datos tiene que seguir el procedimiento especificado en el documento de Implantación de aplicaciones, Capítulo 3.4.1. (Scripts de generación de los objetos de base de datos Oracle).

Se tiene que prestar especial atención a lo siguiente:

- Se tienen que enumerar los archivos (en el caso de instalaciones de muchos archivos es difícil encontrar el archivo a ejecutar) con el número correspondiente al orden de ejecución de la instalación (ejemplo: *01-taules.sql*, *02-vistes.sql*, *03-procediments.sql*, ...).
- Los nombres de archivo no tienen que incluir espacios en blanco ni caracteres especiales.
- No hay que mezclar sentencias DDL y DML en el mismo archivo
- Se ha de evitar enviar archivos muy extensos, ya que si falla algún punto del archivo, éste se continúa ejecutando hasta el final, y puede hacer que la vuelta atrás de la instalación sea muy complicada.
- En el caso de tener que ejecutar *scripts* en lugar de sentencias DML independientes, se tendrá que enviar cada *script* en un archivo a parte.
- Los scripts de base de datos se ejecutan en SQL\*Plus. Es habitual que los desarrolladores utilicen TOAD u otros programas para desarrollar, pero al enviar los scripts de base de datos estos no funcionan en el SQL\*Plus. El error más habitual es no poner la barra (/) al final de los objetos que tienen código (disparadores, scripts con *declare begin end*), ya que estos no se compilan ni ejecutan. La barra (/) se tiene que poner después de cada objeto que contiene código. Por ejemplo, si en un mismo archivo se envían cinco disparadores, tiene que haber una barra (/) después de la definición de cada disparador.
- No se tienen que enviar los scripts DML con sentencias *commit*. Si la ejecución del cuaderno de carga es correcta, los administradores de la base de datos ya harán el *commit* correspondiente. Esto puede evitar muchos problemas ya que si se envían cinco *scripts* y todos hacen *commit*, pero el último de ellos falla, al haberse hecho *commit* en los scripts anteriores no es posible volver atrás la instalación completa haciendo un *rollback*.